

REMARKS

The present application was filed on October 31, 2000 with claims 1-31. In the outstanding Office Action, the Examiner: (i) rejected claims 1-6, 9-15, 20-24 and 29-31 under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,457,797 to Butterworth et al. (hereinafter "Butterworth"); (ii) rejected claims 7, 8, 19 and 28 under 35 U.S.C. § 103(a) as being unpatentable over Butterworth in view of U.S. Patent No. 6,345,278 to Hitchcock et al. (hereinafter "Hitchcock"); (iii) rejected claims 16 and 25 under 35 U.S.C. § 103(a) as being unpatentable over Butterworth in view of U.S. Patent No. 5,922,044 to Banthia (hereinafter "Banthia"); and (iv) rejected claims 17, 18, 26 and 27 under 35 U.S.C. § 103(a) as being patentable over Butterworth in view of U.S. Patent No. 6,275,790 to Yamamoto et al. (hereinafter "Yamamoto").

In this response, Applicants: (i) file a Request for Continued Examination with a Supplemental Information Disclosure Statement; and (ii) traverse the § 102(b) and § 103(a) rejections for at least the following reasons.

Regarding the § 102(b) rejection of claims 1-6, 9-15, 20-24 and 29-31 based on Butterworth, Applicants again respectfully assert that Butterworth fails to teach or suggest all of the limitations in claims 1-6, 9-15, 20-24 and 29-31, for at least the reasons presented below.

It is well-established law that a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). Applicants assert that the rejection based on Butterworth does not meet this basic legal requirement, as will be explained below.

The present invention, for example, as recited in independent claim 1, recites a method for use in a client/server environment of generating a user-interactive application that is dynamically partitionable when deployed in the client/server environment. The method comprises specifying that access to a model associated with the user-interactive application be performed through an application programming interface permitting location-independent allocation and access of model storage on the client and the server. Further, the method comprises specifying that access to view generating logic associated with the user-interactive application be performed through an application

programming interface permitting location-independent allocation and access of view elements on the client and the server. Independent claim 11, 20, 29, 30 and 31 recite respective aspects of the invention having similar limitations.

The claimed invention provides significant advantages over existing partitioning approaches. As illustratively explained in the present specification, at pages 2 and 3:

Whatever the partitioning choice, currently, the choice must be made either in the application's design phase or, at the latest, during application implementation. Controller code is typically "location-dependent" meaning that it is explicitly written to execute on the client or on the server. Even when the Controller is not location-dependent, current View technologies strongly determine a static partitioning of the application. For example, in the current state of the art, because an application's View components are tightly coupled with the manner in which the components are rendered, the choice of a given View technology strongly determines where code involving those components must execute. Thus, if the View consists of HTML components rendered in a web browser, the View components will typically reside on the server because it is non-trivial for a web browser to include code that generates HTML. Alternatively, if the View components are elements of Sun Microsystems™ Swing libraries, the components must reside on the client because Swing components require local display media.

A similar situation applies with regard to the application's Model. Typically, the Model is statically partitioned because the data must physically reside either on the client or on the server. For example, allocating Model storage on either the client-side or the server-side of the application ties Controllers that access that Model to execute on only that side of the application.

Thus, advantageously, the claimed invention provides a technique for generating a user-interactive application that is dynamically partitionable when deployed in the client/server environment which comprises specifying that access to a model associated with the user-interactive application be performed through an application programming interface permitting location-independent allocation and access of model storage on the client and the server, and specifying that access to view generating logic associated with the user-interactive application be performed through an application programming interface permitting location-independent allocation and access of view elements on the client and the server.

Furthermore, as pointed out in the background section of the present specification:

Deployment to multiple platforms from a single-source View description is not new, e.g., see IBM Corporation™ VisualAge OpenClass libraries and Sun Microsystem™ Java AWT . . . which already provide similar functionality.

The IBM Corporation™ VisualAge for Java Ultra Light Client service offering . . . provides server-side Swing-compliant View components which are transparently rendered on the client side by a lightweight client application. In contrast, as will be explained below, the present invention may use actual Swing components (in one embodiment) on the client side, and corresponding non-visual server-side components that can be converted to and from the client-side components. (Underling added for emphasis).

Butterworth discloses a technique for multi-platform partitioning for computer applications. As mentioned above (citing the background section of the present specification), such multi-platform partitioning is known. However, Butterworth fails to teach or suggest various limitations of the claimed invention. By way of example, Butterworth fails to teach or suggest generating a user-interactive application that is dynamically partitionable when deployed in a client/server environment which specifies that access to view generating logic associated with the user-interactive application be performed through an application programming interface permitting location-independent allocation and access of view elements on the client and the server, as in the claimed invention.

The present Office Action cites various portions of Butterworth in support of the rejection. However, not only do none of these parts (nor any parts) of Butterworth teach or suggest “an application programming interface permitting location-independent allocation and access of view elements on the client and the server,” as in the claimed invention, but none of these parts (nor any parts) of Butterworth even teach or suggest “access to view generating logic associated with the user-interactive application be performed through an application programming interface,” as in the claimed invention. This is because Butterworth does not contemplate dynamic placement of view elements through “an application programming interface permitting location-independent allocation and access of view elements on the client and the server,” as in the claimed invention.

More particularly, in the present Office Action, the Examiner cites column 6, line 54, in combination with column 5, lines 36-41, column 6, lines 42-49, column 9, lines 33-35, column 22, lines 10-16 and 23-25, of Butterworth in rejecting the claimed feature of “an application

programming interface permitting location-independent allocation and access of view elements on the client and the server.” However, the only cited portion of Butterworth that mentions the phrase “application programming interface” is column 6, line 54, which mentions “an API to the New York Stock Exchange.” It is not clear how this cited portion of Butterworth in view of the other cited portions of Butterworth could teach or suggest the claimed feature of “an application programming interface permitting location-independent allocation and access of view elements on the client and the server.”

The present Office Action, at page 11, also gives a definition of an application programming interface, and then concludes that an “API is a code or program, which is not a patentable limitation.” Besides this statement, on its face, being legally incorrect, Applicants point out that they are not attempting to patent the concept of a generic API without regard to its specific functionalities. However, Applicants respectfully believe that they are at least entitled to a claim including the feature of “an application programming interface permitting location-independent allocation and access of view elements on the client and the server.”

For at least the above reasons, Applicants assert that independent claims 1, 11, 20, 29, 30 and 31 are patentable over Butterworth. Furthermore, Applicants assert that claims 2-6, 9, 10, 12-15 and 21-24 are patentable over Butterworth not only due to their respective dependence on independent claims 1, 11 and 20, but also because such claims recite patentable subject matter in their own right.

Regarding the §103 rejection of claims 7, 8, 19 and 28 based on the Butterworth/Hitchcock combination, Applicants again respectfully assert that the cited combination fails to teach or suggest all of the limitations in claims 7, 8, 19 and 28 for at least the reasons presented below. Hitchcock fails to remedy any of the deficiencies presented above with respect to Butterworth. Applicants also assert that dependent claims 7, 8, 19 and 28 are patentable over the cited combination not only due to their respective dependence on independent claims 1, 11 and 20, but also because such claims recite patentable subject matter in their own right.

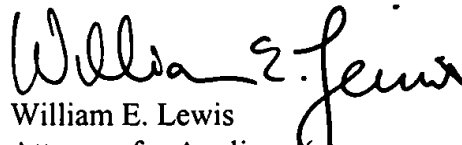
Regarding the §103 rejection of claims 16 and 25 based on the Butterworth/Banthia combination, Applicants again respectfully assert that the cited combination fails to teach or suggest all of the limitations in claims 16 and 25 for at least the reasons presented below. Banthia fails to

remedy any of the deficiencies presented above with respect to Butterworth. Applicants also assert that dependent claims 16 and 25 are patentable over the cited combination not only due to their respective dependence on independent claims 11 and 20, but also because such claims recite patentable subject matter in their own right.

Regarding the §103 rejection of claims 17, 18, 26 and 27 based on the Butterworth/Yamamoto combination, Applicants again respectfully assert that the cited combination is improper under 35 U.S.C. §103(c). In accordance with §103(c), Applicants assert that the present application and Yamamoto were, at the time the invention of the present application was made, owned by (or at least subject to an obligation of assignment to) International Business Machines Corporation of Armonk, NY. Thus, Yamamoto is disqualified as a reference. It appears that the present Office Action fails to address this issue, despite the fact that it was raised by Applicants in their previous response.

In view of the above, Applicants believe that claims 1-31 are in condition for allowance, and respectfully request withdrawal of the §102(b) and §103(a) rejections.

Respectfully submitted,



William E. Lewis

Attorney for Applicant(s)

Reg. No. 39,274

Ryan, Mason & Lewis, LLP

90 Forest Avenue

Locust Valley, NY 11560

(516) 759-2946

Date: August 16, 2004